PCT/EP2004/ 13-01-05

BUNDESREPUBLIK DEUTSCHLAND

EP04/14832

Prioritätsbescheinigung über die Einreichung einer Patentanmeldung

Aktenzeichen: 10 2004 001 435.3

Anmeldetag: 09. Januar 2004

Anmelder/Inhaber: Elektro Beckhoff GmbH Unternehmensbereich Industrie Elektronik, 33415 Verl/DE

Bezeichnung: Verfahren, Schnittstelle und Netzwerk zum zyklischen Versenden von Ethernet-Telegrammen

IPC: H 04 L 12/56

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 7. Januar 2005
Deutsches Patent- und Markenamt
Der Präsident
Im Auftrag

Hintermeier

A 9161
03/00
EDV-L

Verfahren, Schnittstelle und Netzwerk zum zyklischen Versenden von Ethernet-Telegrammen

Die Erfindung betrifft ein Verfahren, eine Schnittstelle und ein Netzwerk zum zyklischen Versenden von Ethernet-Telegrammen.

Das Ethernet ist die am weitesten verbreitete Technologie, mit der in lokalen Kommunikationsnetzen, sog. Local Area Networks (LAN), Daten aktuell mit einer Geschwindigkeit bis zu 100 Mio. Bits/s (Mbps) übertragen werden können. LANs sind lokale Kommunikationsnetzwerke, die auf ein geografisches Gebiet begrenzt sind und sich aus einem oder mehreren Servern und Arbeitsstationen, sog. Knoten zusammensetzen, die über eine Übertragungsstrecke, z.B. ein Koaxial-, Glasfaser- oder Twisted Pair-Kabel verbunden sind. Bei LANs sind verschiedenste Netzwerktopologien möglich, wobei die bekanntesten die Bus-, Ring-, Stern- oder Baumstrukturen sind.

LANs werden mit einem Netzwerk-Betriebssystem und einem einheitlichen Netzwerk-Protokoll betrieben. Das Ethernet stellt ein mögliches Netzwerkprotokoll dar und unterstützt dabei die unterschiedlichsten Kommunikationsprotokolle, z.B. das TCP/IP-Protokoll oder das IPX-Protokoll. Im OSI-Schichtenmodell, dem internationalen Referenzmodell für Datenübertragung in Netzwerken, das aus einem Schichtenstapel aus sieben Schichten aufgebaut ist, wobei für jede Schicht eine Menge von Protokollen definiert ist, die jeweils der nächst höheren Schicht ihre Dienste zur Verfügung stellen, ist das Ethernet- der zweiten Schicht, der sog. Leitungsschicht zugeordnet. In dieser Leitungsschicht werden die zu versendeten Daten zu Telegrammen gebündelt, denen spezifische Informationen für das jeweilige Kommunikationsprotokoll hinzugefügt werden. Die Leitungsschicht ist im Netzwerk für den Transport der Datentelegrammme von Knoten zu Knoten und für die Fehlererkennung zuständig.

Beim Ethernet-Konzept ist die Leitungsschicht in zwei Ebenen unterteilt, wobei die erste Ebene den Daten einen Kopfabschnitt, einen sog. Startkennung hinzufügen, der Informationen enthält, die für eine korrekte Datenübertragung vom Empfängerprotokoll benötigt werden. In der zweiten Ebene des Ethernet-Protokolls wird dann das Datentelegramm mithilfe einer zusätzlichen Präambel und eines Endabschnitts, einer sog. Checksumme, für den Transport von Knoten zu Knoten eingekapselt. Mit solchen Ethernet-Telegrammen lassen sich Daten mit einer Länge von bis zu 1500 Bytes übertragen, wobei zwischen den einzelnen Ethernet-Telegrammen eine feste Pausenzeit einzuhalten ist.

Für das Versenden und das Empfangen der Ethernet-Telegramme auf der Ethernet-Übertragungsstrecke ist ein Ethernet-Controller, auch als Media Access Controller (MAC) bezeichnet, zuständig, der zwischen den Knoten und die Ethernet-Übertragungsstrecke geschaltet und über ein Bussystem mit dem Knoten verbunden ist. Dieser Ethernet-Controller wird in der Regel durch einen Software-Treiber, der im jeweiligen Betriebssystem des Knotens eingebunden ist, gesteuert. Der Ethernet-Controller umfasst im Allgemeinen ein Sende- und ein Empfangs-Schieberegister, um die Ethernet-Übertragungsstrecke von dem physikalischen Speicher des Knotens zu entkoppeln. Moderne Ethernet-Controller besitzen weiter in der Regel eine direkte Zugriffsmöglichkeit auf den physikalischen Speicher des Knotens, einen sog. Direct Memory Access (DMA), wodurch der Software-Treiber im Betriebssystem des Knotens zeitsparend die zu sendenden und zu empfangenen Ethernet-Telegramme direkt im Speicher des Knotens ablegen bzw. aus diesem Speicher abholen kann.

Ethernet-Protokolle werden vornehmlich bei Bürokommunikationsnetzwerken eingesetzt. Aufgrund der Vorteile des Ethernet-Konzepts bei der Nutzung von Standard-Hard- und -softwarekomponenten sowie der Möglichkeit, bei einfacher Vernetzungstechnologie hohe Datenübertragungsraten zu erreichen, werden

Ethernet-Kommunikationsnetzwerke zunehmend auch in der industriellen Fertigung zum Datenaustausch zwischen Arbeitsstationen einzusetzen. Beim Einsatz des Ethernet-Protokolls in der Automatisierungstechnik muss jedoch mithilfe zusätzlicher aufwändiger Hard- und/oder Softwaretechniken die Echtzeitfähigkeit der Ethernet-Datenübertragung sichergestellt werden. Bei der Steuerung von Maschinen ist es in der Regel erforderlich, dass eine zyklische Bearbeitung einer Steuerungsaufgabe im Wesentlichen ohne zeitliche Schwankungen, d.h. sog. Jitter erfolgt, wobei mit einer vorhersehbaren Antwortzeit auf die Regelanforderung reagiert wird.

Sollen z. B. im Rahmen einer auf einem als Knoten in eine Ethernet-Netzwerk ausgebildeten Steuerungsrechner laufenden Echtzeitanwendung zyklisch Ethernet-Telegramme versendet werden, um per Ethernet-Übertragungsstrecke angebundene Sensoren und Aktoren anzusprechen, übergibt der Steuerungsrechner über den im Betriebssystem integrierten Software-Treiber in jedem Steuerungszyklus entsprechende Ethernet-Telegramme an seinen Ethernet-Controller zum Versenden. Dabei fügt der Software-Treiber vor der Übergabe an den Ethernet-Controller den zu versendenden Echtzeitdaten automatisch die in der Ethernet-Übertragungsnorm (IEEE 802.3) definierten Pausenzeiten, Startkennungen, Präambeln und Checksummen hinzu. Der Ethernet-Controller lädt dann, vorzugsweise mithilfe der Direct Memory Access-Übertragung, die entsprechenden Ethernet-Telegramme in seinen Sende-Schieberegister und beginnt ab einem bestimmten Füllstand des Sende-Schieberegisters mit dem Versenden der Ethernet-Telegramme auf der Ethernet-Übertragungsstrecke.

In dieser Sende-Abfolge des Steuerungsrechners mit angeschlossenem Ethernet-Controller sind mehrere Jitter-behaftete Vorgänge enthalten, deren Jitter sich im ungünstigsten Fall aufsummieren und dann einen maximal zulässigen Wert für die Echtzeitanwendung, der in der Regel im Bereich von einigen wenigen Mikrosekunden liegt, übersteigt. Zum Jitter tragen

dabei die schwankenden Interrupt-Latenzzeiten im Knoten beim Erzeugen der zu versendenden Daten und die Laufzeitschwankungen des bis zum Senden des Ethernet-Telegramms durchlaufenden Programmcodes bei. Bei modernen Steuerungsrechnern, die über einen Cache-Speicher verfügen, schwankt zusätzlich auch die Laufzeit der Programmcodes, da je nach Cache-Inhalt unterschiedlich lange auf den angeforderten Inhalt des Speichers im Knoten gewartet werden muss.

Der Ethernet-Controller ist in der Regel über ein Bussystem mit dem Knoten verbunden, wobei häufig ein PCI-Bus verwendet wird. Da ein solcher Bus im Allgemeinen auch von anderen Systemteilen genutzt wird, kann es bei der Buszuteilung zu unterschiedlich langen Wartezeiten kommen. Dies gilt sowohl dann, wenn der Ethernet-Controller per Direct Memory Access-Übertragung auf den physikalischen Speicher des Steuerungsrechners zugreift, als auch für den Fall, dass die Echtzeitdaten unter Kontrolle des Software-Treibers über das Bussystem übertragen werden. Es treten immer ähnliche Jitter bei der Buszuteilung auf. Der Ethernet-Controller beginnt darüber hinaus immer erst ab einem bestimmten Füllstand des Sende-Schieberegisters mit dem Versenden der Ethernet-Telegramme auf der Ethernet-Übertragungsstrecke. Dabei kann sich dann das Senden der Ethernet-Telegramme je nach Füllstand des Sende-Schieberegisters unterschiedlich lange verzögern, was zu einem zusätzlichen Jitter führt.

Ist der sich ergebende Gesamtjitter beim Sendevorgang höher als für die jeweilige Echtzeitanwendungen maximal zulässig Jitter, muss eine solche Abweichung mithilfe eines aufwändigen Verfahrens, wie z.B. dem IEEE 1588 (IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems) eine entsprechend genaue Zeitbasis in allen Kommunikationsteilnehmern an der Ethernet-Übertragungsstrecke geschaffen werden, mit deren Hilfe sich dann der Jitter kompensieren lässt.

Aufgabe der Erfindung ist es, ein Verfahren zum Versenden von Daten in Form von Ethernet-Telegrammen auf einer Ethernet-Übertragungsstrecke, eine Schnittstelle zum Anbinden eines Knotens an eine Ethernet-Übertragungsstrecke und ein Ethernet-Netzwerk bereitzustellen, mit denen sich auf einfache Weise jitterfrei und zyklisch Ethernet-Telegramme, insbesondere mit Echtzeitdaten versenden lassen.

Diese Aufgabe wird mit einem Verfahren nach Anspruch 1, einer Schnittstelle nach Anspruch 9 und einem Ethernet-Netzwerk nach Anspruch 14 gelöst. Bevorzugte Weiterbildungen sind in den abhängigen Ansprüchen angegeben.

Erfindungsgemäß wird zum Versenden von Daten in Form von Ethernet-Telegrammen auf einer Ethernet-Übertragungsstrecke mit Hilfe einer Schnittstelle zum Anbinden eines Knotens an die Ethernet-Übertragungsstrecke die zu versendenden Daten mithilfe einer Umsetzeinheit gemäß einer Übertragungsnorm des Ethernet-Protokolls umgesetzt, um Ethernet-Telegramme bereitzustellen, und mithilfe einer Sendeeinheit dann fortlaufend die bereitgestellten Ethernet-Telegramme versendet, wobei kontinuierliche Ethernet-Telegramme auf die Ethernet-Übertragungsstrecke ausgegeben werden.

Durch das erfindungsgemäße kontinuierliche Senden von Ethernet-Telegrammen wird eine exakte Wiederholbarkeit des Sendevorgangs und damit ein jitterfreies Versenden der Ethernet-Telegramme ermöglicht. Dadurch, dass die Schnittstelle zur Anbindung des Knotens an das Ethernet im Anschluss an ein gesendetes Ethernet-Telegramm direkt das nächste Ethernet-Telegramm versendet, wird gewährleistet, dass alle jitterbehafteten Vorgänge in der Sendeabfolge vom Umsetzen der zu versendenden Daten in Ethernet-Telegramme bis zur Ausgabe der Telegramme auf die Ethernet-Übertragungsstrecke kompensiert werden. Das Timing des Sendevorgangs wird nämlich ausschließlich durch die Schnittstelle des Knotens zum Ethernet be-

stimmt, wobei durch das kontinuierliche Versenden der Telegramme deren völlige Jitterfreiheit sichergestellt wird.

Gemäß einer bevorzugten Ausführungsform der Erfindung wird zum fortlaufenden Senden der bereitgestellten Ethernet-Telegramme bei einer vorgebenen Länge der Ethernet-Telegramme die Länge der Zykluszeit im Rahmen der maximal zulässigen Dauer des Zyklus angepasst, um während der gesamten Zykluszeit kontinuierlich Ethernet-Telegramme auf die Ethernet-Übertragungsstrecke auszugeben. Diese Vorgehensweise sorgt für einen kontinuierlichen Sendevorgang der Ethernet-Telegrammen im Rahmen einer vorgebenen maximalen Sendezyklusdauer bei gleichzeitig optimaler Nutzung der Zykluslänge, wobei der bei der Erzeugung der Ethernet-Telegramme auftretende Jitter völlig ausgeglichen wird.

Gemäß einer weiteren bevorzugten Ausführungsform der Erfindung wird beim fortlaufenden Versenden der Ethernet-Telegramme mithilfe der Schnittstelle die Anzahl und/oder die Länge der in einem Zyklus zu versendenden Ethernet-Telegramme an eine vorgegebene Zykluszeit so angepasst, dass während der gesamten vorgegebenen Zykluszeit kontinuierlich Ethernet-Telegramme auf die Ethernet-Übertragungsstrecke ausgegeben werden können. Diese Vorgehensweise ermöglicht einen kontinuierlichen Sendevorgang der Ethernet-Telegrammen im Rahmen eines Sendezyklus bei gleichzeitig optimaler Nutzung der im Zyklus möglichen Datenbreite, wobei der bei der Erzeugung der Ethernet-Telegramme auftretende Jitter völlig ausgeglichen wird.

Gemäß einer weiteren bevorzugten Ausführungsform werden bei der Anpassung der Ethernet-Telegramme die Baudrate der Ethernet-Übertragungsstrecke, die Länge der beim Umsetzen der Daten gemäß der Übertragungsnorm des Ethernet-Protokolls jeweils in das Ethernet-Telegramm eingefügten Startkennung, Präambel und Checksumme und die Länge der zwischen den zu versendenden Ethernet-Telegrammen einzuhaltenden Pausenzeiten

berücksichtigt. Durch diese Vorgehensweise lässt sich auf einfache Weise die optimale Länge der kontinuierlich auf der Ethernet-Übertragungsstrecke zu versendenden Ethernet-Telegramme bestimmen. Bevorzugt ist dabei weiterhin, dass berücksichtigt wird, dass dann, wenn die berechnete Länge des Ethernet-Telegramms größer als die maximal mögliche Länge des Ethernet-Telegramms ist, die Anzahl und die Länge der zu sendenden Ethernet-Telegramme so gewählt wird, dass in einem Zyklus mehrere vorzugsweise gleich lange Ethernet-Telegramme versendet werden, deren gemeinsame Bitlänge der Zykluszeit entspricht. Hierdurch wird sichergestellt, dass auf einfache Weise eine optimale Länge der zu versendenden Ethernet-Telegramme bestimmt werden kann.

Gemäß einer weiteren bevorzugten Ausführungsform werden die bereitgestellten Ethernet-Telegramme in einem Zwischenspeicher abgelegt, wobei die Schnittstelle den Sendevorgang in Abhängigkeit eines vorgegebenen Füllstandes im Zwischenspeicher startet. Diese Vorgehensweise sorgt dafür, dass immer genügend zu sendende Ethernet-Telegramme in der Schnittstelle vorhanden sind, um einen kontinuierlichen Sendevorgang zu gewährleisten. Hierdurch wird verhindert, dass der Sendevorgang leerläuft und es dadurch zu Telegrammverzögerungen kommt, die dann zu einer Zyklusverletzung führen würden.

Erfindungsgemäß ist weiter vorgesehen, dass dann, wenn die Daten Echtzeitdaten sind, eine die zu versendenden Echtzeitdaten erzeugende Echtzeitanwendung im Knoten mit dem Sendevorgang der Ethernet-Telegramme synchronisiert ist. Diese Vorgehensweise verhindert einen Überlauf mit Ethernet-Telegrammen in der Schnittstelle beim Sendevorgang, der dazu führen würde, dass die Ethernet-Telegramme nicht mehr schnell genug versendet werden können. Dadurch, dass die im Knoten ablaufende Echtzeitanwendung auf die Zeitbasis der den Sendevorgang ausführenden Schnittstelle abgestimmt ist, wird gewährleistet, dass der Knoten Ethernet-Telegramm nur abge-

stimmt mit dem Sendevorgang der Schnittstelle liefert und somit kein Überlauf an Ethernet-Telegrammen auftritt.

Gemäß einer weiteren bevorzugten Ausführungsform ist das Ethernet-Netzwerk mit der Ethernet-Übertragungsstrecke, an der eine Mehrzahl von Knoten angebunden sind, so ausgelegt, dass die Ethernet-Telegramme auf dem Sendekanal kollisionsfrei übertragen werden können. Hierdurch wird gewährleistet, dass auf dem Ethernet kontinuierlich gesendet werden kann, ohne dass es aufgrund von Kollisionen auf der Übertragungsstrecke zu einer Unterbrechung des Sendevorgangs und damit zu einer Zyklusverletzung kommt.

Bevorzugt ist dabei weiterhin, dass die Ethernet-Übertragungsstrecke eine ringförmig angeordnete Topologie aufweist, wobei die vom Sendeknoten versandten Ethernet-Telegramme von einem Knoten zum nächsten weitergeleitet werden. Diese Vorgehensweise ermöglicht eine kollisionsfreie Übertragung der Ethernet-Telegramme mit geringer Verzögerung von einem Knoten zum nächsten.

Die Erfindung wird anhand der beigefügten Zeichnungen näher erläutert. Es zeigen:
Fig. 1A ein Ethernet-Netzwerk;
Fig. 1B einen erfindungsgemäßen Auslegung eines Knoten-Anschlusses im Ethernet-Netzwerk;
Fig. 2A ein Ethernet-Telegramm und
Fig. 2B einen erfindungsgemäßen Sendevorgang.

Das Ethernet-Konzept ist der am weitesten verbreitete Kommunikationsstandard in lokal begrenzten Kommunikationsnetzwerken (LAN), über die sich auf einfache Weise Datenressourcen zwischen Arbeitsstationen, im Allgemeinen Computer oder Maschinen, im Weiteren auch als Knoten bezeichnet, gemeinsam nutzen lassen. Das Ethernet basiert dabei auf einem LAN-Aufbau, bei dem eine Mehrzahl von Knoten über ein gemeinsames Übertragungsmedium miteinander verbunden sind, wobei das E-

thernet-Konzept die Verkapselung der zu übermittelnden Daten in Datenpaketen, im Weiteren auch als Ethernet-Telegramme bezeichnet, mit einem vorbestimmten Format vornimmt. Das Ethernet besteht dabei aus drei Bereichen, nämlich der Übertragungsstrecke und den Netzwerk-Schnittstellen, also der Hardware, der Menge an Protokollen, die den Zugriff auf die Ethernet-Übertragungsstrecke steuern, und dem Ethernet-Telegrammformat.

Fig. 1A zeigt schematisch ein Ethernet-Netzwerk, bei dem mehrere Knoten 1 über eine Ethernet-Übertragungsstrecke 2 miteinander verbunden sind. Die Anbindung des Knotens an die Ethernet-Übertragungsstrecke erfolgt dabei mithilfe eines Ethernet-Controllers 3, der vorzugsweise in den zugehörigen Knoten integriert ist. Ein erfindungsgemäßer Knoten 1 mit angeschlossenem Ethernet-Controller 3 zur Anbindung an die Ethernet-Übertragungsstrecke 2 ist im Detail in Fig. 1B gezeigt. Für das Versenden der Ethernet-Telegramme im Ethernet-Controller ist eine Codiereinheit 31 zuständig, für den Empfang der Ethernet-Telegramme von der Übertragungsstrecke 2 eine Decodiereinheit 32. An die Codiereinheit 31 bzw. die Decodiereinheit 32 ist jeweils ein als Schieberegister ausgelegter Zwischenspeicher 33, 34 angeschlossen, um die zu versendenden bzw. empfangenen Ethernet-Telegramme zwischenzuspeichern. Diese Sende- und Empfangs-Schieberegister 33, 34 wiederum sind vorzugsweise so ausgelegt, dass sie auf einen physikalischen Speicher 11 im Knoten 1 direkt mithilfe des sogenannten Direct Memory Access (DMA) Modus zugreifen können. Alternativ besteht die Möglichkeit, dass der Datenaustausch zwischen dem Sende-Schiebespeicher 33 bzw. dem Empfangs-Schiebespeicher 34 und dem physikalischen Speicher 11 über eine zentrale Recheneinheit (CPU) 12 des Knotens 1 erfolgt. Der direkte Zugriff über den DMA-Modus ermöglicht jedoch einen beschleunigten Datenaustausch.

Die Steuerung des Datenaustausches zwischen dem physikalischen Speicher 11 des Knotens 1 und der Schnittstelle 2 er-

folgt in der Regel durch die CPU 12 des Knotens 1. Die CPU 12 des Knotens verwaltet weiter auch alle zum Betrieb des Ethernet notwendigen Vorgänge, d.h. führt das Management des Sende- und Empfangsvorgangs durch und sorgt für die Verkapselung der zu versendenden Daten des Knotens in Ethernet-Telegramme bzw. das Entpacken der Daten aus den empfangenen Ethernet-Telegrammen. Das auf der CPU 12 des Knotens 1 implementierte Betriebssystem weist in der Regel eine geschichtete Softwarestruktur auf, um eine protokollspezifische Bearbeitung von einer telegramm- und hardwarespezifischen Bearbeitung zu trennen. Dadurch ist es möglich, unterschiedliche Kommunikationsprotokolle beim Ethernet-Standard einzusetzen, ohne am hardwarespezifischen Treiber Änderungen durchführen zu müssen. Gleichzeitig kann dann auch die Hardware des Knotens geändert werden, ohne gleichzeitig eine protokollspezifische Softwareänderung ausführen zu müssen.

Ein Ethernet-Telegramm 5, dessen Struktur in Fig. 2A schematisch gezeigt ist, kann bis zu 1500 Bytes enthalten und setzt sich aus einem Kopfteil mit einer Startkennung 51, einer Präambel 52, die die Ziel- und Quelladresse und den Datenpakettyp kennzeichnet, einem Mittelteil 53 mit Daten und einem eine Checksumme enthaltenden Endteil 54, die als Fehlerkennungsmechanismus dient, zusammen.

Ein Sendevorgang von Ethernet-Telegramme über die Ethernet-Übertragungsstrecke 2 wird so ausgeführt, dass der in der CPU 12 eingesetzte Software-Treiber die zu versendenden Daten in Ethernet-Telegramme umsetzt, die, wenn der Ethernet-Controller 3 im DMA-Modus arbeitet, im physikalischen Speicher 11 des Knotens 1 abgelegt werden. Auf diese abgespeicherten Ethernet-Telegramme greift dann das Sende-Schieberegister 33 des Ethernet-Controllers 3 zu, um die Ethernet-Telegramme in das Schieberegister zu laden. Wenn unter Steuerung durch den Software-Treiber in der CPU 12 vom physikalischen Speicher 11 genügend Ethernet-Telegramme auf das Sende-Schieberegister 33 übertragen wurden und damit ein

ausreichender Füllstand erreicht ist, gibt das Sende-Schieberegister 33 die zwischengespeicherten Ethernet-Telegramme über die Codiereinheit 31 auf die Ethernet-Übertragungsstrecke 2 aus. Eine Ethernet-Datenübertragung findet dabei nur dann statt, wenn das Ethernet-Netzwerk ruhig ist. Darüber hinaus ist in der Regel zusätzlich ein Kollisionsverhinderungsmechanismus auf der Ethernet-Übertragungsstrecke 2 vorgesehen.

Beim Empfang von Ethernet-Telegrammen werden die empfangenen Ethernet-Telegramme von der Decodiereinheit 32 im Empfangs-Schieberegister 34 zwischengespeichert, wobei der Ethernet-Controller 3 einen Interrupt im Knoten 1 auslöst. Dieser Interrupt veranlasst den Software-Treiber der CPU 12 im Knoten 1, die empfangenen Telegramme über den DMA-Modus in den physikalischen Speicher 11 überzuführen und dann an das Betriebssystem im Knoten zur Verarbeitung weiterzuleiten.

Das Ethernet-Konzept wird vor allem deshalb als Kommunikationsprotokoll für Netzwerksysteme genutzt, da Standard-Hardware- und -softwarekomponenten verwendet werden können und darüber hinaus eine hohe Datenübertragungsrate möglich ist. Beim Einsatz des Ethernet-Standards in industrieller Umgebung, insbesondere für Automatisierungsaufgaben, muss das Ethernet-Protokoll aber eine Echtzeit-Datenübertragung gewährleisten. Um zuverlässig eine Echtzeitanwendung, wie z. B. eine Maschinensteuerung mithilfe eines Ethernet-Netzwerkes ausführen zu können, ist ein Datenaustausch mit Zykluszeiten von 50 µsec bei zulässigen Jitterzeiten, d.h. Abweichungen von der gewünschten Zykluszeit von 10 µsec erforderlich.

Wenn ein Steuerungsrechner, der einen Knoten 1 in einem Ethernet-Netzwerk darstellt, Sensoren oder Aktoren, die als weitere Knoten an die Ethernet-Übertragungsstrecke 2 angebunden sind, in Echtzeit steuern soll, werden vom Steuerungsrechner in jeden Steuerungszyklus mithilfe des in seiner CPU 12 eingespeicherten Software-Treibers Ethernet-Telegramme 5

an den zugehörigen Ethernet-Controller 3 zum Senden übergeben. Der Ethernet-Controller 3 wird dann die entsprechenden Ethernet-Telegramme 5, vorzugsweise per DMA-Modus in sein Sende-Schieberegister 33 laden und ab einem bestimmten Füllstand dieses Senderegisters mit dem Versenden der Ethernet-Telegramme auf der Ethernet-Übertragungsstrecke 2 beginnen.

In dieser Sendeabfolge sind jedoch mehrere jitterbehaftete Vorgänge enthalten, deren Jitter sich im ungünstigsten Fall aufsummiert. Ein erster Jitter ergibt sich bereits aus den schwankenden Interrupt-Latenzzeiten des Betriebssystems des Steuerungsrechners und des Software-Treibers beim Umsetzen der Ethernet-Telegramme. Weiterhin treten Laufzeitschwankungen des bis zum Versenden des Ethernet-Telegrammen durchlaufenden Datencodes auf. Bei modernen Steuerungsrechnern, die über einen Cache-Speicher verfügen, schwanken zusätzlich auch die Laufzeiten ein und desselben durchlaufenden Datencodes, da je nach Cache-Inhalt unterschiedlich lange auf den angeforderten Speicher gewartet werden muss. Weitere Jitter ergeben sich zudem bei der Übergabe der Ethernet-Telegramme an den Ethernet-Controller. Der Ethernet-Controller ist über ein Bussystem, z.B. einen PCI-Bus, an den Steuerungsrechner, angebunden. Da der Bus auch von anderen Systemteilen des Steuerungsrechners genutzt wird, kann es bei der Buszuteilung zu unterschiedlich langen Wartezeiten beim Zugriff des Ethernet-Controllers auf den physikalischen Speicher zur Übergabe der Ethernet-Telegramme an das Sende-Schieberegister kommen. Auch dann, wenn der Ethernet-Controller nicht per DMA-Übertragungsmodus arbeitet, sondern die Daten über die CPU aus dem physikalischen Speicher in das Sende-Schieberegister des Ethernet-Controllers übertragen werden, treten ähnliche Jitter bei der Buszuteilung auf. Weiterhin wird das Senden der Ethernet-Telegramme je nach Füllstand der Sende-Schieberegister unterschiedlich lang verzögert. Wenn sich alle genannten jitterbehafteten Vorgänge aufsummieren, besteht die Gefahr, dass der sich ergebende Gesamtjitter höher als

der für die jeweilige Echtzeitanwendung zulässige Jitter ist und dann keine Echtzeitsteuerung mehr gewährleistet wird.

Um nicht, wie herkömmlicherweise aufwändige Verfahren zur Angleichung der Zeitbasis zwischen den einzelnen Knoten und damit zur Kompensation der Kommunikationsjitter ausführen zu müssen, wird erfindungsgemäß der Ethernet-Controller 3 über den Software-Treiber auf der CPU 12 des Rechner-Knotens 1 so programmiert, dass ohne Pause Ethernet-Telegramme 5 aus dem Sende-Schieberegister 33 versendet werden. Dabei wird das Sende-Schieberegister 33 und die angeschlossene Codiereinheit 31 des Ethernet-Controllers 3 so gesteuert, dass im Anschluss an ein gesendetes Ethernet-Telegramm direkt das nächste Ethernet-Telegramm unter Einhaltung der in der Ethernet-Übertragungsnorm definierten Pausezeit gesendet wird.

Um zu gewährleisten, dass bei einer vorgegebenen Zykluszeit der auszuführenden Echtzeitanwendung kontinuierlich Ethernet-Telegramme versendet werden, berechnet der Software-Treiber der CPU 12, wie viele und wie lange Ethernet-Telegramme versendet werden müssen, um die vorgegebene Zykluszeit exakt einzuhalten. Der Softwaretreiber stellt die zu versendenden Daten 53 gemäß der Ethernet-Übertragungsnorm in entsprechend lange Ethernet-Telegramme 5 mit Startkennung 51, Präambel 52 und Checksumme 54 zusammen und legt sie im physikalischen Speicher 11 des Knotens 1 ab. Das Sende-Schieberegister 33 des Ethernet-Controllers 3 greift dann auf diese Ethernet-Telegramme 5 zu und speichert sie zwischen. Ab einem gewissen Füllstand im Sende-Schieberegister 33 wird dann mit dem Sendevorgang begonnen, wobei kontinuierlich Ethernet-Telegramme versendet werden, wie in Fig. 2B gezeigt ist. Hier ist ein Sendevorgang dargestellt, bei dem innerhalb einer vorgegebenen Zykluszeit zwei gleich lange Telegramme unter Einhaltung der vorgeschriebenen Pausenzeit versendet werden.

Mithilfe des im Ethernet-Controller 3 integrierten Sende-Schieberegisters 33 wird die Bereitstellung der Ethernet-

Telegramme durch den Software-Treiber der CPU 13 im physikalischen Speicher 11 des Knotens 1 vom Sendezeitpunkt dieser Ethernet-Telegramme entkoppelt, so dass der bei der Echtzeitanwendung und die bei der Übertragung der Ethernet-Telegramme auf den Ethernet-Controller 3 entstehenden Jitter ausgeglichen werden. Da das Timing des Sendevorgangs ausschließlich vom Ethernet-Controller 3 und der nachgeschalteten Übertragungsphysik der Ethernet-Übertragungsstrecke 2 abhängt und der Ethernet-Controller aus seinem Sende-Schieberegister 33 Ethernet-Telegramme 5 kontinuierlich sendet, ist eine exakte Wiederholbarkeit und damit ein jitterfreies Senden möglich.

Um ein kontinuierliches Senden der Ethernet-Telegramme zu ermöglichen, ist die Echtzeitanwendung im Knoten über den Software-Treiber der CPU 12 auf den Ethernet-Controller 3 synchronisiert. Der Ethernet-Controller 3 legt die Zeitbasis fest, auf die die Echtzeitanwendung auf dem Steuerungsrechner 1 synchronisiert ist. Dadurch wird sichergestellt, dass vom Software-Treiber der CPU 12 immer genügend zu sendende Ethernet-Telegramme an den Ethernet-Controller 3 übergeben werden, um zu verhindern, dass das Sende-Schieberegister 33 des Ethernet-Controllers 3 leerläuft, und so Telegrammverzögerungen auftreten, die zu einer Zykluszeitverletzung führen würden. Weiterhin wird durch die Synchronisierung der Echtzeitanwendung im Knoten 1 auf die Zeitbasis des Ethernet-Controllers 3 gewährleistet, dass nicht zu viele Ethernet-Telegramme an den Ethernet-Controller 3 übergeben werden und somit ein Überlauf im Sende-Schieberegister 33 stattfindet und die Ethernet-Telegramme nicht mehr schnell genug versendet werden können.

Bei der Berechnung der Anzahl und der Länge in einem Zyklus der Echtzeitanwendung zu versendenden Ethernet-Telegramme berücksichtigt der Software-Treiber der CPU 12 im Knoten 1 sowohl die auf der Ethernet-Übertragungsstrecke 2 verwendete Baudrate als auch die bei der Verkapselung der zu versendenden Daten automatisch eingefügten zusätzlichen Daten, d.h.

Startkennung 51, Präambel 52 und Checksumme 54, sowie die zwischen den Ethernet-Telegrammen einzuhaltenden Pausenzeiten. Diese zusätzlichen Signale sind in der Ethernet-Norm IEEE 802.3 festgelegt und betragen bei einem 100-Base-TX-Ethernet, also einem Fast Ethernet mit 100 MBaud für die Startkennung 8 Bit, für die Präambel 56 Bit, für die Checksumme 32 Bit und die Pausenzeit 69 Bit.

Soll nun eine Zykluszeit der Echtzeitanwendung von x µsec erreicht werden, dann gilt folgende Formel: (L ist die maximale Bitlänge des Ethernet-Telegramms)

$$L = (x \cdot 100) - (8 + 56 + 32 + 69)$$

Für eine Zykluszeit von 100 µsec ergibt sich dann:

$$L = 9808 \text{ Bits} = 1226 \text{ Bytes}.$$

Der Software-Treiber der CPU 12 im Knoten 1 kann also in einer Zykluszeit von 100 µsec ein oder auch mehrere gleich lange Telegramme mit einer Gesamtlänge einschließlich der Pausenzeiten von 1226 Bytes versenden. Wenn z.B. zwei Telegramme versendet werden, so beträgt die Bytelänge dann 613 Bytes pro Telegramm (einschließlich Pausezeit). Eine Aufteilung in mehrere Telegramme ist dann zwingend erforderlich, wenn die bei einer Zykluszeit sich ergebende Telegrammlänge größer als die maximale Ethernet-Telegramme von 1500 Byte ist. Es müssen dann immer entsprechend mehrere Ethernet-Telegramme zur Nutzung der vollen Zykluszeit gesendet werden. Wird z.B. eine Zykluszeit von 500 µsec vorgegeben, können fünf Telegramme mit je 1226 Byte (100 µsec) gesendet werden.

Alternativ zu Vorgabe einer Zykluszeit besteht auch die Möglichkeit, um einen kontinuierlichen Versand von Ethernet-Telegramme zu erreichen, die Länge der zu versendenden Ethernet-Telegramme vorzugeben, um daraus dann die notwendige Zykluszeit abzuleiten. Der Software-Treiber der CPU berechnet in

diesem Fall aus der Vorgabe der Länge der Ethernet-Telegramme und der maximal zulässigen Dauer des Steuerzyklus, um eine Echtzeitanwendung, wie z. B. eine Maschinensteuerung mithilfe der Ethernet-Netzwerkes ausführen zu können, die optimale Zykluszeit, die einen kontinuierlichen Versand der Ethernet-Telegramme gewährleistet. Der Softwaretreiber stellt dann wiederum die zu versendenden Daten gemäß der Ethernet-Übertragungsnorm in entsprechend lange Ethernet-Telegramme mit Startkennung 51, Präambel 52 und Checksumme 54 zusammen und legt sie im physikalischen Speicher 11 des Knotens 1 ab. Das Sende-Schieberegister 33 des Ethernet-Controllers 3 greift anschließend auf diese Ethernet-Telegramme 5 zu und speichert sie zwischen. Ab einem gewissen Füllstand im Sende-Schieberegister 33 wird dann mit dem Sendevorgang begonnen, wobei kontinuierlich Ethernet-Telegramme innerhalb der berechneten Zykluszeit unter Einhaltung der vorgeschriebenen Pausenzeit versendet werden.

Beim erfindungsgemäßen Sendevorgang, bei dem quasi kontinuierlich gesendet wird, muss weiter gewährleistet sein, dass auf dem Sendekanal keine Kollisionen auftreten, da dann der Ethernet-Controller die Übertragung unterbricht und erst später wieder aufnimmt muss. Eine geeignete Topologie des Ethernet-Netzwerkes für einen kollisionsfreien Sendevorgang stellt dabei eine Peer-zu-Peer-Verbindung zwischen den Knoten dar. Es besteht auch die Möglichkeit, mehrere Teilnehmer über einen Switch, der Kollisionen verhindert, anzusteuern. Auch eine ringförmige Netztopologie von mehreren Knoten ist möglich, wobei dann die Ethernet-Telegramme mit geringer Verzögerung von Knoten zu Knoten weitergeleitet und anschließend an den ursprünglichen Sende-Knoten zurückgeschickt werden.

Echtzeitanwendungen benötigen in der Regel auch Rückmeldungen der angesteuerten Teilnehmer. In diesem Fall ist die Ethernet-Übertragungsstrecke 2 als Vollduplex-Übertragungsstrecke mit getrenntem Sende- und Empfangskanal ausgebildet, um die zu versendenden Ethernet-Telegramme nicht durch empfangene

Telegramme mit den Rückmeldungen zu beeinflussen. Die Menge der zurückgesendeten Daten darf dabei auch nicht die Menge der ursprünglich gesendeten Daten überschreiten, da diese dem Maximum der Übertragungskapazität entsprechen.

Patentansprüche

1. Verfahren zum Versenden von Daten in Form von Ethernet-Telegrammen auf einer Ethernet-Übertragungsstrecke, mit den Verfahrenschritten
Umsetzen der zu versendenden Daten gemäß einer Übertragungsnorm des Ethernet-Protokolls, um Ethernet-Telegramme zu bereitzustellen, und
fortlaufendes Senden der bereitgestellten Ethernet-Telegramme, wobei kontinuierlich Ethernet-Telegramme auf die Ethernet-Übertragungsstrecke ausgegeben werden.

2. Verfahren nach Anspruch 1, wobei zum fortlaufenden Senden der bereitgestellten Ethernet-Telegramme bei einer vorgebenen Länge der Ethernet-Telegramme die Länge der Zykluszeit im Rahmen der maximal zulässigen Dauer des Zyklus angepasst wird, um während der gesamten Zykluszeit kontinuierlich Ethernet-Telegramme auf die Ethernet-Übertragungsstrecke auszugeben.

3. Verfahren nach Anspruch 1, wobei zum fortlaufenden Senden der bereitgestellten Ethernet-Telegramme bei einer vorgegebenen Zykluszeit die Anzahl und/oder die Länge der in einem Zyklus zu versendenden Ethernet-Telegramme angepasst wird, um während der gesamten vorgebenen Zykluszeit kontinuierlich Ethernet-Telegramme auf die Ethernet-Übertragungsstrecke ausgeben.

4. Verfahren nach Anspruch 3, wobei zum Berechnen der Anzahl und/oder der Länge der in einem Zyklus zu versendenden Ethernet-Telegramme die auf der Ethernet-Übertragungsstrecke verwendete Baudrate, die Länge der beim Umsetzen der Daten gemäß der Übertragungsnorm des Ethernet-Protokolls jeweils in das Ethernet-Telegramm eingefügten Startkennung, Präambel und Checksumme und die Länge der zwischen den zu versendeten Ethernet-Telegrammen einzuhaltenden Pausenzeit berücksichtigt werden.

5. Verfahren nach Anspruch 4, wobei die maximale Bitlänge L des in einem Zyklus zu versendenden Ethernet-Telegramms bei einer auf der Ethernet-Übertragungsstrecke verwendeten Baudrate von ba Mbaud, einer Zykluszeit von zy µs, einer Startkennungslänge von st Bit, einer Präambellänge von pr Bit, einer Checksummenlänge von ch Bit und einer Pausenzeit von pa Bit wie folgt berechnet wird:
L = (ba * zy) - (st + pr + ch + pa).

6. Verfahren nach Anspruch 5, wobei dann, wenn die maximale Bitlänge L größer als die maximal mögliche Bitlänge der Ethernet-Telegramme ist, die Anzahl und die Länge der zusendeten Ethernet-Telegramme so gewählt wird, dass in einem Zyklus mehrere Ethernet-Telegramme versendet werden, deren gemeinsame Bitlänge der Zykluszeit entspricht.

7. Verfahren nach einem der Ansprüche 1 bis 6, wobei zum fortlaufenden Senden der bereitgestellten Ethernet-Telegramme die bereitgestellten Telegramme in einem Zwischenspeicher zwischengespeichert werden und der Sendevorgang gestartet wird, sobald eine vorgegebener Füllstand im Zwischenspeicher erreicht ist.

8. Verfahren nach einem der Ansprüche 1 bis 7, wobei die zu versendeten Daten Echtzeitdaten sind und eine die zu versendenden Echtzeitdaten erzeugende Echtzeitanwendung mit dem Sendevorgang der Ethernet-Telegramme synchronisiert wird.

9. Schnittstelle zum Anbinden eines Knotens an ein Ethernet-Übertragungsstrecke (2) mit einer Umsetzeinheit (12) zum Umsetzen von zu versendenden Daten des Knotens gemäß einer Übertragungsnorm des Ethernet-Protokolls, um Ethernet-Telegramme zu bereitzustellen, und
einer Sendeeinheit (31, 33) zum fortlaufenden Versenden der bereitgestellten Ethernet-Telegramme, um kontinuierlich E-

thernet-Telegramme auf die Ethernet-Übertragungsstrecke auszugeben.

10. Verfahren nach Anspruch 9, mit einer Einheit (12) zum Anpassen der Länge der Zykluszeit bei einer vorgebenen Länge der zu versendenden Ethernet-Telegramme im Rahmen der maximal zulässigen Dauer des Zyklus, um während der gesamten Zykluszeit kontinuierlich Ethernet-Telegramme auf die Ethernet-Übertragungsstrecke (2) auszugeben.

11. Schnittstelle nach Anspruch 9, mit einer Einheit (12) zum Anpassen der Anzahl und/oder der Länge der in einem Zyklus zu versendenden Ethernet-Telegramme an eine vorgegebene Zykluszeit, um während der gesamten vorgebenen Zykluszeit kontinuierlich Ethernet-Telegramme auf die Ethernet-Übertragungsstrecke (2) auszugeben.

12. Schnittstelle nach einem der Ansprüche 9 bis 11, mit einem Zwischenspeicher (33) zum Zwischenspeichern der bereitgestellten Ethernet-Telegramme und einer Einheit (12) zum Starten des Sendevorgangs in Abhängigkeit von einem vorgegebenen Füllstand im Zwischenspeicher.

13. Schnittstelle nach einem der Ansprüche 9 bis 12, mit einer Einheit (12) zum Synchronisieren einer die zu versendenden Echtzeitdaten erzeugende Echtzeitanwendung mit dem Sendevorgang der Ethernet-Telegramme.

14. Ethernet-Netzwerk mit Ethernet-Übertragungsstrecke und einer Mehrzahl von Knoten, wobei wenigstens einer der Knoten über eine Schnittstelle nach einem der Ansprüche 9 bis 13 mit der Ethernet-Übertragungsstrecke verbunden sind, wobei der Sendekanal der Ethernet-Übertragungsstrecke ausgelegt ist, die Ethernet-Telegramme kollisionsfrei zu übertragen.

15. Ethernet-Netzwerk nach Anspruch 14, wobei die Ethernet-Übertragungsstrecke eine ringförmig angeordnete Topologie

aufweist und die vom Sende-Knoten versandten Ethernet-Telegramme von einem Knoten zum nächsten Knoten weitergeleitet werden.

Zusammenfassung

Verfahren, Schnittstelle und Netzwerk zum zyklischen Versenden von Ethernet-Telegrammen

Beim Versenden von Daten in Form von Ethernet-Telegrammen 5 auf einer Ethernet-Übertragungsstrecke 2 mit Hilfe einer Schnittstelle 2 zum Anbinden eines Knotens 1 an die Ethernet-Übertragungsstrecke werden die zu versendenden Daten mittels einer Umsetzeinheit 12 gemäß einer Übertragungsnorm des Ethernet-Protokolls umgesetzt, um Ethernet-Telegramme bereitzustellen, und mithilfe einer Sendeeinheit 31, 33 fortlaufend die bereitgestellten Ethernet-Telegramme versendet, wobei kontinuierliche Ethernet-Telegramme auf die Ethernet-Übertragungsstrecke ausgegeben werden.

Fig. 2B

Figur für die Zusammenfassung

Fig. 2B

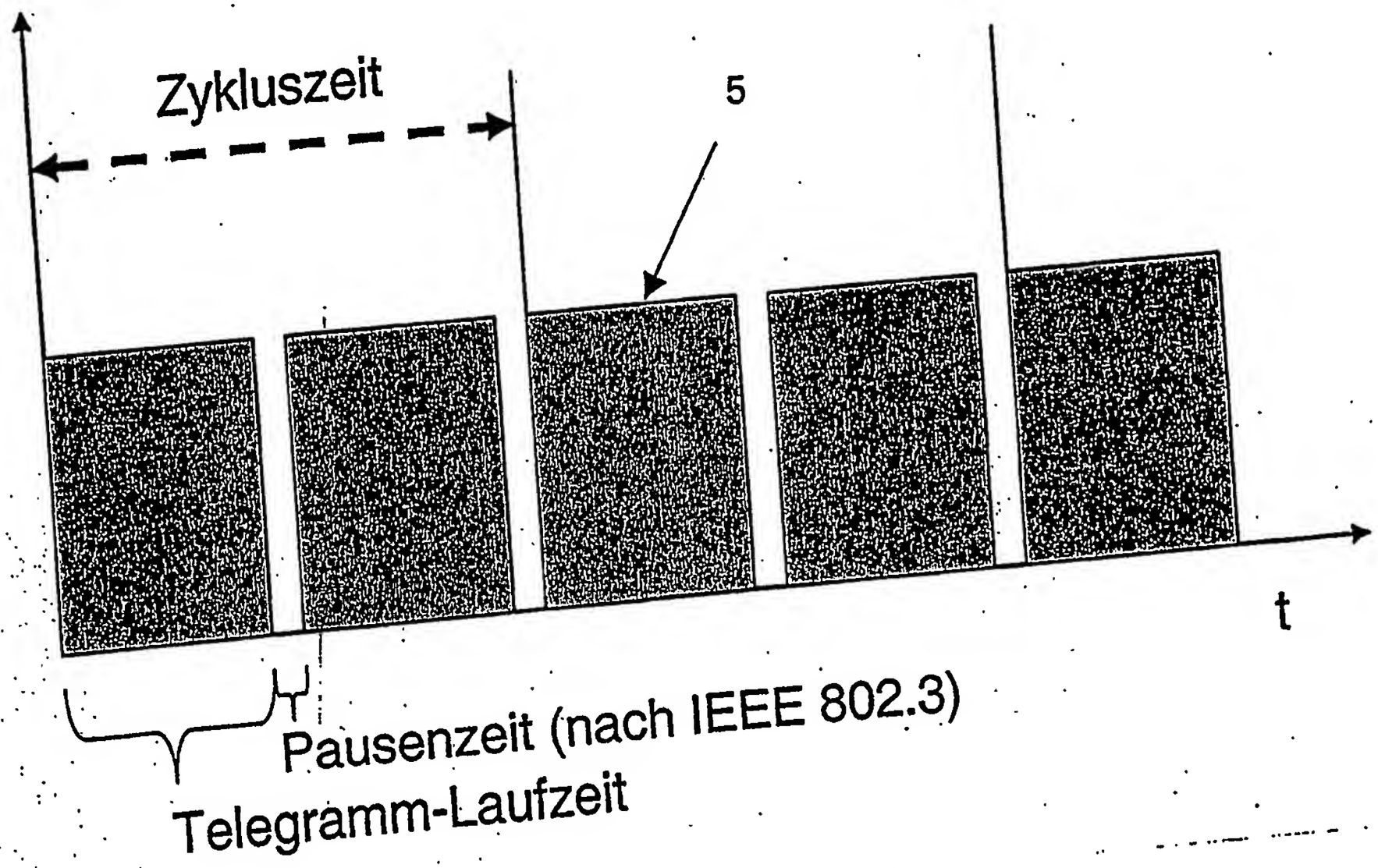


Fig. 1A

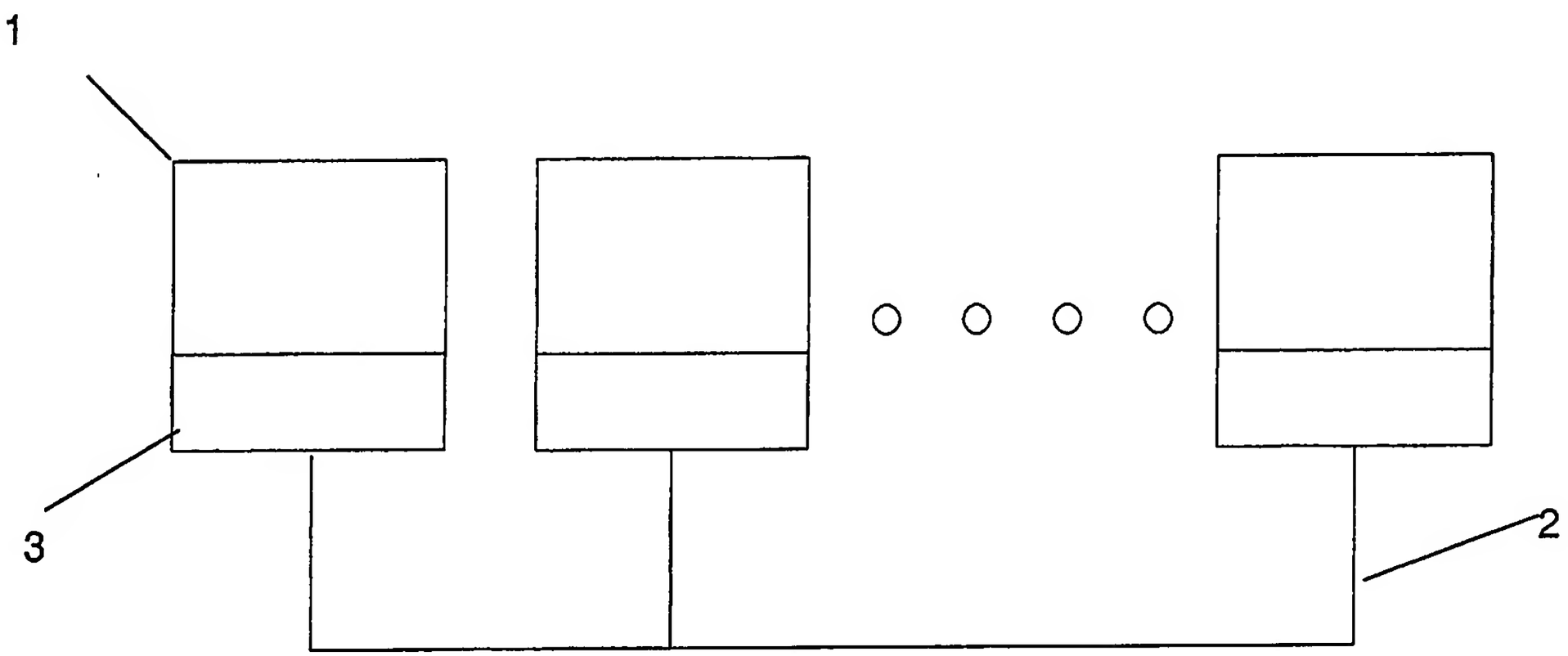


Fig. 1B

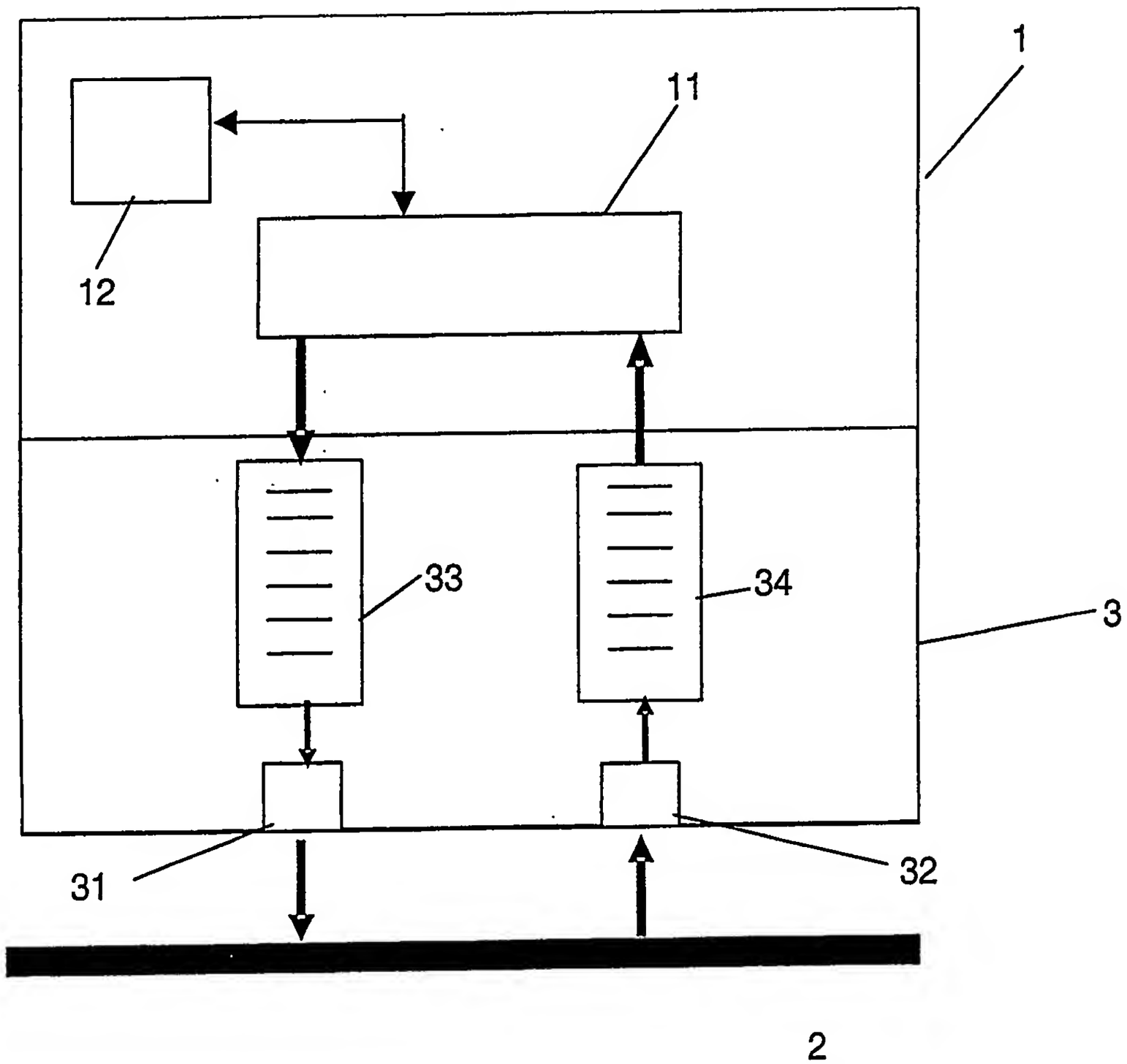


Fig. 2A

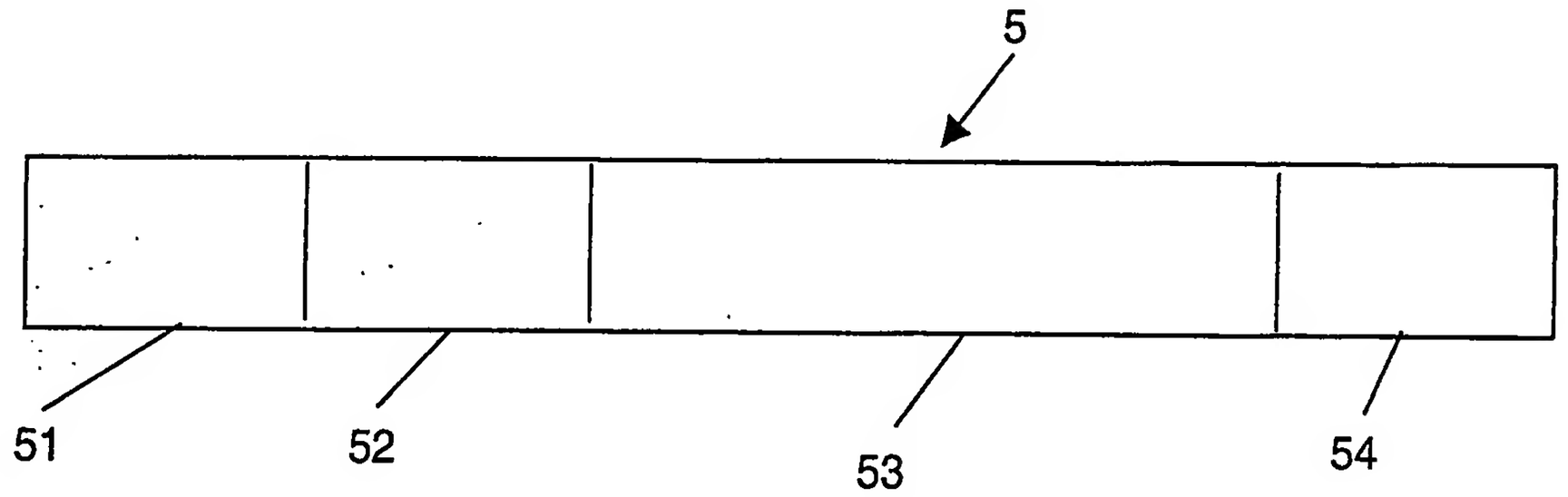
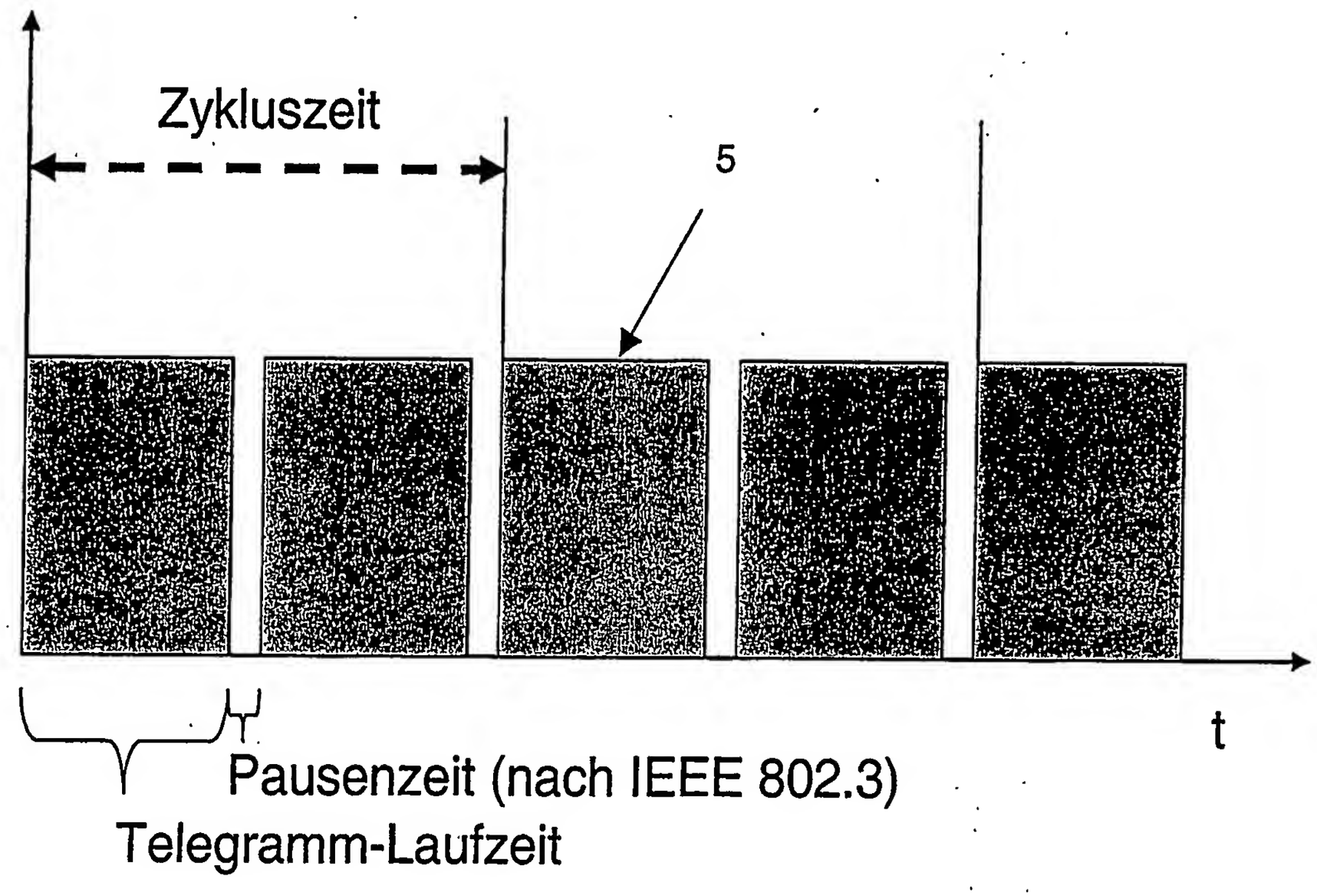

5
51
52
53
54

Fig. 2B


Zykluszeit
5
t
Pausenzeit (nach IEEE 802.3)
Telegramm-Laufzeit

Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/EP04/014832

International filing date: 30 December 2004 (30.12.2004)

Document type: Certified copy of priority document

Document details: Country/Office: DE
Number: 10 2004 001 435.3
Filing date: 09 January 2004 (09.01.2004)

Date of receipt at the International Bureau: 27 January 2005 (27.01.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)

World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse